WEEK - 15

Aim: to implement a YOLO (You only look once) deep learning model for real time object detection images (or) videos.

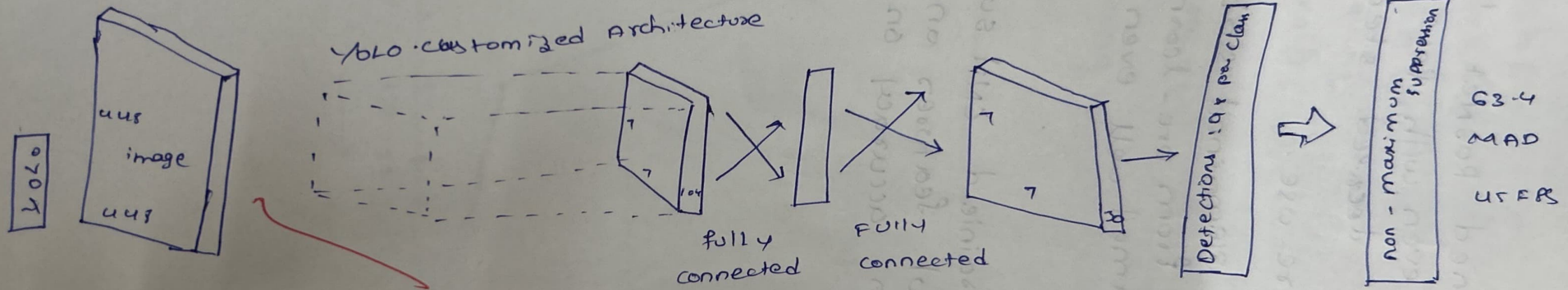
Algorithm:

1) load a pre trained YOLO model
2) Load an input image (or) video frame
3) Resize and normalize image for model input
4) pass image through YOLO Network
5) Get predictions → bounding boxes, Confidence cores, and class labels.
6) Apply Non-maximum suppression (NMS) to remove duplicate boxes
7) Draw boxes and labels on the original image
8) Display detected objects.

Pseudo code:

```
Import YOLO library
load pre-trained YOLO model and weights
Read input image (or) video..
Resize and Normalize image.
forward pass through YOLO model
For each detection:
    Extract confidence score and class id
    if confidence > threshold:
        compute bounding box coordinates
        Apply non-maximum suppression
```

YOLO ARCHITECTURE



Draw rectangle and lable on image
Display output image with detections.

Observation :

→ YOLO detects multiple objects in a single pass efficiently
→ Each object has a bounding box, label & confidence score
→ Accuracy increases when lighting and angle are clear ~~real-ti~~
→ Real-time detection works at high FPS

Result:

the YOLO model succesfully detected and labeled multiple objects.

3/11/25

Output :

448x640 (0 detections), 68.1 ms

Speed: 10.7 ms preprocess, 68.1 ms inference

0.8ms postprocess pre image at shape (1,3,448,640)

```python
from ultralytics import YOLO

# Load a pre-trained YOLOv8 model (e.g., 'n' for nano, 's' for small, etc.)
model = YOLO("yolov8n.pt")

# Perform object detection on an image
# 'source' can be a file path, URL, or even a webcam (source=0)
results = model("path/to/your/image.jpg")
# The 'results' object contains all the detection data: bounding boxes, class, and confidence scores.

# To view the results with bounding boxes drawn, you can save the annotated image:
for result in results:
    # Save the annotated image to a file in the 'runs/detect/predict' directory
    result.save(filename="detected_image.jpg")

print("Detection complete. Annotated image saved.")
```